Introduction to Milestones in Interactive Theorem Proving

Avigad, Jeremy; Blanchette, Jasmin Christian; Klein, Gerwin; Paulson, Lawrence; Popescu, Andrei; Snelting, Gregor

published in
Journal of Automated Reasoning
2018

DOI (link to publisher)
10.1007/s10817-018-9465-5

document version
Publisher's PDF, also known as Version of record

J Autom Reasoning (2018) 61:1–8
https://doi.org/10.1007/s10817-018-9465-5

Introduction to *Milestones in Interactive Theorem Proving*

Jeremy Avigad[1] · **Jasmin Christian Blanchette**[2] · **Gerwin Klein**[3] · **Lawrence Paulson**[4] · **Andrei Popescu**[5] · **Gregor Snelting**[6]

Received: 25 April 2018 / Accepted: 27 April 2018 / Published online: 11 May 2018

On March 8, 2018, Tobias Nipkow celebrated his sixtieth birthday. In anticipation of the occasion, in January 2016, two of his former students, Gerwin Klein and Jasmin Blanchette, and one of his former postdocs, Andrei Popescu, approached the editorial board of the *Journal of Automated Reasoning* with a proposal to publish a surprise Festschrift issue in his honor. The e-mail was sent to twenty-six members of the board, leaving out one, for reasons that will become clear in a moment. It is a sign of the love and respect that Tobias commands from his colleagues that within two days every recipient of the e-mail had responded favorably and enthusiastically to the proposal.

✉ Jeremy Avigad
avigad@cmu.edu

Jasmin Christian Blanchette
j.c.blanchette@vu.nl

Gerwin Klein
gerwin.klein@data61.csiro.au

Lawrence Paulson
lp15@cam.ac.uk

Andrei Popescu
a.popescu@mdx.ac.uk

Gregor Snelting
gregor.snelting@kit.edu

1 Department of Philosophy, Carnegie Mellon University, Pittsburgh, PA, USA

2 Theoretical Computer Science, Vrije Universiteit Amsterdam, Amsterdam, North Holland, Netherlands

3 Trustworthy Systems Research Group, Data61, CSIRO, Sydney, Australia

4 Computer Laboratory, University of Cambridge, Cambridge, UK

5 School of Science and Technology, Middlesex University London, London, UK

6 Department of Computer Science, Karlsruhe Institute of Technology, Karlsruhe, Germany

There were only two problems that had to be addressed, and the subsequent discussion focused on them. The first problem was that Tobias—the one member of the editorial board who was omitted from the mailing list—was actually the editor-in-chief of the journal. He was (and still is) the one tasked with negotiating page limits with the publisher, and the journal contract stipulates that he has to approve all content. Springer assured us that there was absolutely no way of publishing an issue of the journal behind Tobias's back.

The second problem was that Tobias is highly skeptical of Festschriften, and it seemed that half the editorial board could tell stories of him railing against the poor quality of Festschrift articles. In Tobias's view, what the world needs is quality research, not nostalgic academics praising their friends. It would have been ironic (though amusing) to subject him to the very thing he had complained so bitterly about in the past.

In the end, we came up with a much better plan. Klein, together with two of Tobias's many friends on the editorial board, Jeremy Avigad and Larry Paulson, would propose a special issue on a topic near and dear to his heart. We would make sure it was an offer he could not possibly refuse. We would issue a broad call for proposals and subject every submission to a rigorous evaluation. It would be a special journal issue like any other, but *we* would know that we were doing it for Tobias.

And then, at the last minute, we would write an introduction dedicating the issue to him. We would also ask an old friend and colleague, Gregor Snelting, to help us write a few words about Tobias's career and his research. We can all learn from the experiences of others, and so it seems appropriate to include in this special issue a brief reflection on a remarkable career dedicated to formal methods and interactive theorem proving.

1 Overview of the Contents

The special issue's call for papers began as follows:

> The past few decades have seen major achievements in interactive theorem proving, such as the formalization of deep mathematical theorems and significant bodies of theoretical computer science, as well as the verification of complex software and hardware systems. Too often, these impressive results have been published in abbreviated or fragmentary form in conference proceedings, or not at all. This special issue welcomes full-length papers describing past work not previously published in a journal, along with new developments of any length. Small, self-contained proof pearls and applications of all kinds are also welcome.
>
> This special issue will be devoted to applications of interactive theorem proving in their full variety: formalized mathematics, formalized theory, formalized semantics, formal proofs of hardware or software systems. They can be large or small.

Thirteen papers were accepted for publication. Three focus on formally verified mathematics:

- Grzegorz Bancerek, Czesław Byliński, Adam Grabowski, Artur Korniłowicz, Roman Matuszewski, Adam Naumowicz, and Karol Pąk, "The Role of the Mizar Mathematical Library for Interactive Proof Development in Mizar"
- Yves Bertot, Laurence Rideau, and Laurent Théry, "Distant Decimals of π: Formal Proofs of Some Algorithms Computing Them and Guarantees of Exact Computation"
- Fabian Immler, "A Verified ODE Solver and the Lorenz Attractor"

Four focus on verification of software, programming languages, or systems:

- Thomas Bauereiß, Armando Pesenti Gritti, Andrei Popsecu, and Franco Raimondi, "CoSMed: A Confidentiality-Verified Social Media Platform"
- Hao Chen, Xiongnan (Newman) Wu, Zhong Shao, Joshua Lockerman, and Ronghui Gu, "Toward Compositional Verification of Interruptible OS Kernels and Device Drivers"
- Cornelius Diekmann, Lars Hupel, Julius Michaelis, Maximilian Haslbeck, and Georg Carle, "Verified iptables Firewall Analysis and Verification"
- Andreas Lochbihler, "Mechanising a Type-Safe Model of Multithreaded Java with a Verified Compiler"

Four describe logical methods and tools that support formal verification:

- Jasmin Christian Blanchette, Mathias Fleury, Peter Lammich, and Christoph Weidenbach, "A Verified SAT Solver Framework with Learn, Forget, Restart, and Incrementality"
- Qinxiang Cao, Lennart Beringer, Samuel Gruetter, Josiah Dodds, and Andrew W. Appel, "VST-Floyd: A Separation Logic Tool to Verify Correctness of C Programs"
- Łukasz Czajka and Cezary Kaliszyk, "Hammer for Coq: Automation for Dependent Type Theory"
- Anders Schlichtkrull, "Formalization of the Resolution Calculus for First-Order Logic"

The remaining two deal with algorithms and formal languages:

- Mohammad Abdulaziz, Michael Norrish, and Charles Gretton, "Formally Verified Algorithms for Upper Bounding State Space Diameters"
- Christian Doczkal and Gert Smolka, "Regular Language Representations in the Constructive Type Theory of Coq"

2 Tobias Nipkow and His Research

The state of interactive theorem proving today would be markedly poorer if Tobias had not been a part of it. His contributions range from the theoretical underpinnings of interactive theorem proving to the practical development of the Isabelle system, from automated reasoning to infrastructure for interaction, and from verification of pure mathematics to verification of software.

Tobias studied informatics from 1977 to 1982 at the Technische Universität Darmstadt, which was then called the Technische Hochschule Darmstadt. In his master's thesis, he proved, together with Gerhard Weikum, that *sufficient completeness* is decidable in linear, confluent rewrite systems. This result became his first paper [35]. He then moved to the University of Manchester to work on his PhD under Cliff Jones, and in 1987 published his dissertation, *Behavioural Implementation Concepts for Nondeterministic Data Types*.

After that, he turned his attention from abstract data types to unification. At the time, there was a lot of interest in unification in algebraic structures—for example, unification modulo associativity and commutativity. Together with Ursula Martin, he investigated unification in Boolean rings [12]. Tobias proved that unification is unitary and devised an algorithm that is much more general than earlier algorithms by Boole and Löwenheim. In 1989, he published a comprehensive survey with Martin, "Boolean Unification—the Story so Far" [13], and he generalized a number of results to primal algebras in his 1990 article "Unification in Primal Algebras, Their Powers and Their Varieties" [17].

Tobias also contributed important results to higher-order unification (that is, unification modulo the rules of the lambda-calculus) and higher-order rewriting. These include extensions of higher-order unification to incorporate polymorphism [20] and first-order equational

theories [32], the introduction of higher-order critical pairs [19], and the succinct formulation of a unification algorithm for a subclass of lambda-terms [21], originally due to Miller [14].

Tobias's involvement with Isabelle started in the late 1980s. First at MIT and later at the University of Cambridge, he used Isabelle as the basis for research into the implementation of term rewriting. He found that Isabelle's basic inference mechanisms (including the presence of logical variables and unification) made it easy to implement rewriting. His early papers [15,16,18] demonstrated how his rewriting tactics, in combination with induction, could easily prove the correctness of simple functional programs such as quicksort. He conducted these experiments using Isabelle/FOL (first-order logic). Recognizing the limitations of that formalism and drawing on the work of Mike Gordon for motivation and inspiration, he undertook the critical work needed to support higher-order logic. This seems to require polymorphism, but basic ML-style polymorphism needed to be tamed in the context of Isabelle's logical framework. Tobias's background in unification allowed him to discover what was needed: order-sorted unification, allowing controlled polymorphism [22] and ultimately leading to the type class system [31]. His other contributions from that period include Isabelle's Earley parser.

From Isabelle-91 [30] onward, his influence kept growing, especially once he was established as a professor in Munich. One of Isabelle's greatest assets has always been the strength of its automation. Tobias was responsible for the design of the simplifier, which makes use of a database of equational theorems as rewrite rules. Larry Paulson's *auto* proof method builds on this simplifier and is still a workhorse when it comes to dispelling proof goals automatically. Tobias is also responsible for Isabelle's *arith* proof method, which can be used to dispel goals involving linear equations and inequalities.

As the 1990s progressed, Tobias coordinated a major effort to advance the development of Isabelle/HOL, with notable advances that include recursive datatypes, linear arithmetic, and general recursive function definitions. This period also saw early verification experiments [33], the formalization of 100 pages of a textbook on programming language semantics [23], and a major project to verify the type safety of Java [34].

The 2000s saw these two streams of prover technology and applications continue with the addition of features such as locales [3,8], structured intelligible proofs (Isar) [26,39], and advanced user interface technology [38]. Tobias and his students contributed a formalization of the entire JavaCard ecosystem, including the semantics and type system of the source language [24,37], and a formalization of the JVM and its bytecode verifier [9,25]. He and Gerwin Klein later condensed the essence of this work on Java in Jinja [11], which provided the first formally verified compiler for a Java-like language.

Tobias's work on formalizing the semantics of programming languages, combined with his years of experience teaching semantics to students, led him to wonder whether students could learn programming language semantics better through interactive theorem proving. Based on initial material [23] from the 1990s, he and Klein took on the task of formalizing additional lecture content in Isabelle, including compiler theorems and abstract interpretation, in a form students could understand. Initial test runs in lectures were promising: student performance improved and they seemed more engaged. Tobias was not surprised: a good interactive theorem prover is like a good computer game in that it is addictive and gives immediate feedback. With refinements to the material over the years, this work culminated in the textbook *Concrete Semantics* [29], which serves as an introduction to theorem proving as well as an introduction to the semantics of programming languages.

In parallel, perhaps the most pragmatically important application of interactive theorem proving is software verification, and this became the focus of Tobias's work in the mid-1990s. The Java and JavaCard projects were the first larger projects, and afterwards Tobias's group

was a major part of one of the most ambitious software verification projects at the time, the Verisoft project, which aimed at the formal verification of an entire hardware/software stack, including operating system, compiler, and applications. The work on this project transformed Isabelle into a tool for software verification [1,36] that has had a substantial impact and is still in use today. This work from Tobias's group made possible milestones such as the formal verification of the seL4 microkernel [10], which was carried out by Klein's team in Sydney just after he finished his PhD in Munich.

As early at the 1990s, with his work on the large-scale language formalizations, Tobias was interested in making specifications executable and extracting code from them [4]. This led him to one especially interesting application. Almost as soon as Thomas Hales announced his monumental *Flyspeck* project to fully verify his proof of the Kepler conjecture, Tobias arranged to spend a sabbatical visit with Hales, at the University of Pittsburgh, to work on a key component: the enumeration of a class of combinatorial structures known as *tame graphs* [28]. This was the project's first major success, and it is the only part of the verification that was carried out in Isabelle.

Tobias also pioneered techniques for verifying decision procedures, including procedures for integer and linear arithmetic [27]. Isabelle's framework for verification and code extraction later made it possible to verify realistic software systems. Examples include the online social media platform and the SAT solver reported in two of this special issue's papers. The framework also supports current work on a verified optimized model checker, as well as an effort to not only extract code, but to extract *verified* code [7] using the verified CakeML compiler.

Tobias was among the first to recognize the usefulness of automatic counterexample generation to debug conjectures in interactive theorem proving. His group developed three such tools: *Quickcheck*, *Refute*, and *Nitpick* [5]. He was also one of the first users of Sledgehammer, a bridge that integrates first-order automatic theorem provers to provide one-click proof automation in Isabelle. The tool was initially developed under Paulson's lead at Cambridge, but Tobias committed his and his group's time to evaluating and improving it [6]. Recognizing that a lot is lost in translation when exporting Isabelle/HOL problems to untyped first-order logic, he started a fruitful collaboration with Christoph Weidenbach's group, which develops the SPASS system, aiming at reducing the gap between interactive and automatic theorem provers.

Beyond his research, Tobias has provided remarkable editorial, organizational, and pedagogical service to the formal methods community. This year marks two decades since the publication of his book *Term Rewriting and All That* [2], written jointly with Franz Baader, which combined a thorough introduction to the theory of rewrite systems with original research. It is still one of the most popular monographs in the area. He founded the steering committee for the *Interactive Theorem Proving* (ITP) conference, and as editor-in-chief, he expanded the *Journal of Automated Reasoning* to become the premier venue for research in interactive theorem proving and formal verification, as well as automated reasoning. He is a founding editor of *ACM Transactions on Computational Logic* (TOCL) and *Logical Methods in Computer Science* (LMCS). He is a founding editor of Isabelle's *Archive of Formal Proofs* (AFP), the world's largest and most rapidly growing repository of formalized mathematics and computer science. He has been a frequent co-organizer of the high-profile Marktoberdorf Summer School, which helped shape many of today's formal methods scientists. He has chaired editions of important conferences such as *International Symposium on Formal Methods* (FM), *Interactive Theorem Proving* (ITP), *International Joint Conference on Automated Reasoning* (IJCAR), and *Rewriting Techniques and Applications* (RTA).

Tobias has had a powerful, lasting influence on the people he has worked with. He has had more than a hundred collaborators and twenty-five PhD students. His very nature drives people to do quality work: he has high standards, and we all want to impress him. Time and time again, projects that he has encouraged others to embark on have turned into lifelong passions and pursuits. In addition to his ability to instill high scientific standards in young researchers, Tobias is also known for the great care he devotes to their career development. His mentorship extends well beyond research to foster the skills needed to write grant proposals, navigate administrative minefields, and establish oneself in academe.

Few researchers can boast the range of Tobias's strengths. He is a solid theoretician, but excels at implementation. He works well alone or with a close colleague, but equally well in a large group. He can throw himself into a focused project, but also has the stamina to sustain a long-term effort. He can employ both firmness and diplomacy to get a job done. He has had great success with his own projects, while at the same time inspiring others to launch projects of their own.

It has been a pleasure to survey some of Tobias's accomplishments here. Interactive theorem proving and automated reasoning have benefited immensely from his contributions. More importantly, those of us who have had the good fortune to interact with him over the years are better off for having known him, and for that, we are grateful.

References

1. Alkassar, E., Hillebrand, M.A., Leinenbach, D., Schirmer, N., Starostin, A.: The Verisoft approach to systems verification. In: Shankar, N., Woodcock, J. (eds.) Verified Software: Theories, Tools, Experiments, Second International Conference, VSTTE 2008, Toronto, Canada, October 6–9, 2008. Proceedings, vol. 5295 of Lecture Notes in Computer Science, pp. 209–224. Springer, New York (2008)
2. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press, Cambridge (1998)
3. Ballarin, C.: Locales and locale expressions in Isabelle/Isar. In: Berardi, S., Coppo, M., Damiani, F. (eds.) Types for Proofs and Programs, International Workshop, TYPES 2003, Torino, Italy, April 30–May 4, 2003, Revised Selected Papers, vol. 3085 of Lecture Notes in Computer Science, pp. 34–50. Springer (2003)
4. Berghofer, S., Nipkow, T.: Executing Higher Order Logic. In: Aagaard, M., Harrison, J. (eds.) Types for Proofs and Programs, International Workshop, TYPES 2000, Durham, UK, December 8–12, 2000, Selected Papers, vol. 1869 of Lecture Notes in Computer Science, pp. 24–40. Springer (2000)
5. Blanchette, J.C., Nipkow, T.: Nitpick: a counterexample generator for higher-order logic based on a relational model finder. In: Kaufmann, M., Paulson, L.C. (eds.) Interactive Theorem Proving, First International Conference, ITP 2010, Edinburgh, UK, July 11–14, 2010. Proceedings, vol. 6172 of Lecture Notes in Computer Science, pp. 131–146. Springer (2010)
6. Böhme, S., Nipkow, T.: Sledgehammer: Judgement day. In: Giesl, J., Hähnle, R. (eds.) Automated Reasoning, 5th International Joint Conference, IJCAR 2010, Edinburgh, UK, July 16–19, 2010. Proceedings, vol. 6173 of Lecture Notes in Computer Science, pp. 107–121. Springer (2010)
7. Hupel, L., Nipkow, T.: A verified compiler from Isabelle/HOL to CakeML. In: Ahmed, A. (ed.) Programming Languages and Systems–27th European Symposium on Programming, ESOP 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14–20 (2018)
8. Kammüller, F., Wenzel, M., Paulson, L.C.: Locales—A sectioning concept for Isabelle. In: Bertot, Y., Dowek, G., Hirschowitz, A., Paulin-Mohring, C., Théry, L. (eds.) Theorem Proving in Higher Order Logics, 12th International Conference, TPHOLs'99, Nice, France, September, 1999, Proceedings, vol. 1690 of Lecture Notes in Computer Science, pp. 149–166. Springer (1999)
9. Klein, G.: Verified Java bytecode verification. PhD thesis, Technische Universität München, Germany (2003)
10. Klein, G., Andronick, J., Elphinstone, K., Heiser, G., Cock, D., Derrin, P., Elkaduwe, D., Engelhardt, K., Kolanski, R., Norrish, M., Sewell, T., Tuch, H., Winwood, S.: seL4: formal verification of an operating-system kernel. Commun. ACM **53**(6), 107–115 (2010)
11. Klein, G., Nipkow, T.: A machine-checked model for a Java-like language, virtual machine, and compiler. ACM Trans. Program. Lang. Syst. **28**(4), 619–695 (2006)

12. Martin, U., Nipkow, T.: Unification in Boolean rings. J. Autom. Reason. **4**(4), 381–396 (1988)
13. Martin, U., Nipkow, T.: Boolean unification–the story so far. J. Symb. Comput. **7**(3/4), 275–293 (1989)
14. Miller, D.: A logic programming language with lambda-abstraction, function variables, and simple unification. J. Log. Comput. **1**(4), 497–536 (1991)
15. Nipkow, T.: Equational reasoning in Isabelle. Sci. Comput. Program. **12**, 123–149 (1989)
16. Nipkow, T.: Term rewriting and beyond–theorem proving in Isabelle. Form. Asp. Comput. **1**, 320–338 (1989)
17. Nipkow, T.: Unification in primal algebras, their powers and their varieties. J. ACM **37**(4), 742–776 (1990)
18. Nipkow, T.: Constructive rewriting. Comput. J. **34**, 34–41 (1991)
19. Nipkow, T.: Higher-order critical pairs. In: Proceedings of the Sixth Annual Symposium on Logic in Computer Science (LICS '91), Amsterdam, The Netherlands, July 15–18 (1991)
20. Nipkow, T.: Higher-order unification, polymorphism, and subsorts. In: Kaplan, S., Okada, M. (eds.) Conditional and Typed Rewriting Systems. CTRS 1990, vol. 516, pp. 436–447 (1991)
21. Nipkow, T.: Functional unification of higher-order patterns. In: Proceedings of the Eighth Annual Symposium on Logic in Computer Science (LICS '93), Montreal, Canada, June 19–23, 1993, pp. 64–74. IEEE Computer Society (1993)
22. Nipkow, T.: Order-sorted polymorphism in Isabelle. In: Huet, G., Plotkin, G. (eds.) Logical Environments, pp. 164–188. Cambridge University Press, Cambridge (1993)
23. Nipkow, T.: Winskel is (almost) right: Towards a mechanized semantics textbook. Form. Asp. Comput. **10**(2), 171–186 (1998)
24. Nipkow, T.: Invited talk: Embedding programming languages in theorem provers (abstract). In: Ganzinger, H. (ed.) Automated Deduction—CADE-16, 16th International Conference on Automated Deduction, Trento, Italy, July 7–10, 1999, Proceedings, vol. 1632 of Lecture Notes in Computer Science, pp. 398. Springer (1999)
25. Nipkow, T.: Verified bytecode verifiers. In: Honsell, F., Miculan, M. (eds.) Foundations of Software Science and Computation Structures, 4th International Conference, FOSSACS 2001 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2001 Genova, Italy, April 2–6, 2001, Proceedings, vol. 2030 of Lecture Notes in Computer Science, pp. 347–363. Springer (2001)
26. Nipkow, T.: Structured proofs in Isar/HOL. In: Geuvers, H., Wiedijk, F. (eds.) Types for Proofs and Programs, pp. 259–278. Springer, Berlin (2003)
27. Nipkow, T.: Linear quantifier elimination. J. Autom. Reason. **45**(2), 189–212 (2010)
28. Nipkow, T., Bauer, G., Schultz, P.: Flyspeck I: tame graphs. In: Furbach, U., Shankar, N. (eds.) Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17–20, 2006, Proceedings, vol. 4130 of Lecture Notes in Computer Science, pp. 21–35. Springer (2006)
29. Nipkow, T., Klein, G.: Concrete Semantics-With Isabelle/HOL. Springer, New York (2014)
30. Nipkow, T., Paulson, L.C.: Isabelle-91 (system abstract). In: Kapur, D. (ed.) Automated Deduction—CADE-11 International Conference, LNAI 607, pp. 673–676. Springer (1992)
31. Nipkow, T., Prehofer, C.: Type checking type classes. In: Principles of Programming Languages, POPL '93, pp. 409–418. ACM, New York (1993)
32. Nipkow, T., Qian, Z.: Modular higher-order *E*-unification. In: Book, R.V. (ed.) Rewriting Techniques and Applications, 4th International Conference, RTA-91, Como, Italy, April 10–12, 1991, Proceedings, vol. 488 of Lecture Notes in Computer Science, pp. 200–214. Springer (1991)
33. Nipkow, T., Slind, K.: I/O automata in Isabelle/HOL. In: Dybjer, P., Nordström, B., Smith, J. (eds.) Types for Proofs and Programs: International Workshop TYPES '94 Båstad, Sweden, June 6–10, 1994 Selected Papers, pp. 101–119. Springer (1995)
34. Nipkow, T., von Oheimb, D.: Java$_{\ell ight}$ is type-safe—definitely. In: Principles of Programming Languages, POPL '98, pp. 161–170. ACM (1998)
35. Nipkow, T., Weikum, G.: A decidability result about sufficient-completeness of axiomatically specified abstract data types. In: Cremers, A.B., Kriegel, H.-P. (eds.) Theoretical Computer Science, 6th GI-Conference, Dortmund, Germany, January 5–7, 1983, Proceedings, vol. 145 of Lecture Notes in Computer Science, pp. 257–268. Springer (1983)
36. Schirmer, N.: A verification environment for sequential imperative programs in Isabelle/HOL. In: Baader, F., Voronkov, A. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning, 11th International Conference, LPAR 2004, Montevideo, Uruguay, March 14–18, 2005, Proceedings, vol. 3452 of Lecture Notes in Computer Science, pp. 398–414. Springer (2004)
37. von Oheimb, D.: Analyzing Java in Isabelle/HOL: formalization, type safety and Hoare logic. PhD thesis, Technische Universität München, Germany (2001)
38. Wenzel, M.: Isabelle/jEdit—a prover IDE within the PIDE framework. In: Jeuring, J., Campbell, J.A., Carette, J., Reis, G.D., Sojka, P., Wenzel, M., Sorge, V. (eds.) Intelligent Computer Mathematics—11th International Conference, AISC 2012, 19th Symposium, Calculemus 2012, 5th International Workshop,

DML 2012, 11th International Conference, MKM 2012, Systems and Projects, Held as Part of CICM 2012, Bremen, Germany, July 8–13, 2012. Proceedings, vol. 7362 of Lecture Notes in Computer Science, pp. 468–471. Springer (2012)
39. Wenzel, M.: Isabelle/Isar—a versatile environment for human-readable formal proof documents. PhD thesis, Institut für Informatik, Technische Universität München (2002)